Implementation of ‘Yarn-Level Simulation of Woven Cloth’

Jiahao Wen

August 5, 2019 in Tokyo

1 Introduction

Woven cloth is formed by interlacing yarns. Typically, the two sets of orthogonal yarns are named warp and weft. Interlaced yarns undergo friction forces at yarn-yarn contacts, and this friction holds together the woven fabric. Simulating cloth in yarn-level is an interesting research topic because some visually interesting effects such as detailed tearing, snags, or loose yarn ends can be captured in contrast to traditional cloth simulation using Finite Element Method or Mass and Spring System. However, modeling individual yarns can also introduce very high computational cost due to large amount of DoFs and frequent yarn-yarn contacts.

In this article, I will introduce my implementation of the paper 'Yarn-Level Simulation of Woven Cloth' published on Siggraph Asia 2014. (All the diagrams except results in this article are from this paper.) The key contribution of this paper is that it proposed a discretization form of yarn crossing, which is the most basic unit in woven cloth. And this paper formulated a novel model of woven cloth which enables efficient yarn-level simulation. It should be noted that I have not implemented all the contents in this paper. I have gotten rid of some effects such as shear friction and shear jamming, which have little influence on the correctness of the whole system.

2 Discretization Model

2.1 Yarn Crossing

In woven cloth, the vast majority of yarns are in contact at yarn crossings. One principal assumption is that such contacts are maintained throughout the simulation, which means the relative sliding between warp and weft can be regarded as the crossing node sliding along different directions. We parameterize warp and weft yarns based on their rest arc length, u and v respectively. Then, we describe a yarn crossing by its Lagrangian position $\boldsymbol{x}$ and the parametric coordinates of the warp and weft material points at the crossing. The variation of

u and v coordinates models, respectively, the sliding of warp and weft yarns. A yarn crossing is a 5-DoF node with 3 Lagrangian DoFs and 2 Eulerian DoFs. Denote the $\mathbf{R}^5$ coordinates of the i^{th} yarn crossing node as $\boldsymbol{q}_i = (\boldsymbol{x}_i, u_i, v_i)$.

Thus, a discrete woven cloth is a combination of 5-DoF yarn-crossing nodes and regular 3-DoF Lagrangian nodes. Usually we set 5-DoF crossing node at each crossing inside the cloth and regular 3-DoF Lagrangian node at the end-points of each single yarn on the boundary. Figure 1 illustrates a regular setup with one yarn-crossing node and its 4 neighboring nodes.

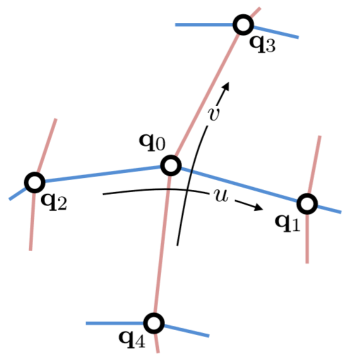

Figure 1: One yarn-crossing node and its 4 neighboring nodes.

The whole system looks kind of similar with mass and spring system. In the geometry model, we ignore the volume of each yarn, and all yarns are initialized flat on the same plane. However, for the purpose of force computation and rendering, the **crimp** caused by interlacing must be considered. Our dealing method is that compute the normal vector of each node in each simulation step, and define that the normal vector points from warp yarn toward weft yarn. When it comes to rendering or computing bending force, we offset each crossing node by the yarn radius R in the corresponding direction to get correct result. We get each node's normal vector by approximating the plane constituted of itself and its neighbors using SVD method.

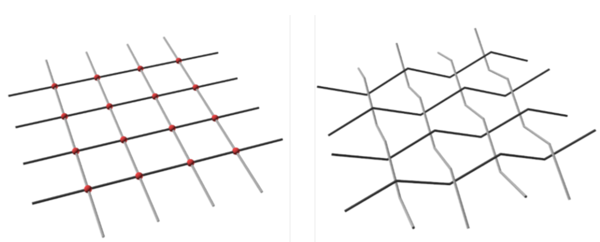

Figure 2: The left figure is the discrete model before offsetting. The right figure is the discrete model after offsetting.

2.2 Motion Equation

In our current implementation, we use the linearly implicit integration scheme at the velocity level, popularized by the work on efficient cloth simulation by Baraff and Witkin in 1998.

$$(M - h^2 K)\dot{q}^{k+1} = M\dot{q}^{(k)} + hf. \tag{1}$$

where h is the time step, and the superscript k indicates the current time step. M is the generalized mass matrix. K is the stiffness matrix. After adding the equality and inequality constraints, which may be introduced by self-collision, rigid body contact, or some interactive operation, we can turn the problem into a quadratic programming problem:

$$\begin{aligned} \underset{\dot{q}}{minimize} \quad & \frac{1}{2}\dot{q}^T \widetilde{M}\dot{q} - \dot{q}^T \widetilde{f} \\ subject\ to \quad & A_{eq}\dot{q} = 0 \\ & A_{ineq}\dot{q} \geq 0 \end{aligned}$$

where $\widetilde{M} = M - h^2 K$, and $\widetilde{f} = M\dot{q}^{(k)} + hf$. In our implementation, we solve this quadratic programming problem by Mosek, which is an academic free software providing C++ library. Finally, both the Lagrangian and Eulerian DoFs are updated by $q^{(k+1)} = q^{(k)} + h\dot{q}^{k+1}$.

2.3 Generalized Mass Matrix

As for the generalized mass matrix of the system, we assume the mass is distributed uniformly along yarns, with density ρ. We can derive that the generalized mass matrix for one warp segment $[\boldsymbol{q}_0, \boldsymbol{q}_1]$ (similar for a weft segment) is

$$\boldsymbol{M}_{0,1} = \frac{1}{6}\rho\Delta u \begin{pmatrix} 2\boldsymbol{I}_3 & -2\boldsymbol{w} & \boldsymbol{I}_3 & -\boldsymbol{w} \\ -2\boldsymbol{w}^T & 2\boldsymbol{w}^T\boldsymbol{w} & -\boldsymbol{w}^T & \boldsymbol{w}^T\boldsymbol{w} \\ \boldsymbol{I}_3 & -\boldsymbol{w} & 2\boldsymbol{I}_3 & -2\boldsymbol{w} \\ -\boldsymbol{w}^T & \boldsymbol{w}^T\boldsymbol{w} & -2\boldsymbol{w}^T & 2\boldsymbol{w}^T\boldsymbol{w} \end{pmatrix} \tag{2}$$

where $\boldsymbol{w} = \frac{\boldsymbol{x}_1 - \boldsymbol{x}_0}{\Delta u}$, and $\Delta u = u_1 - u_0$. The derivation can be referred in the original paper. In the implementation, we compute the local mass matrix for each segment and fill them in the global mass matrix $\boldsymbol{M}$.

3 Force Model

The forces in the system include external forces (e.g. gravity) and internal forces caused by woven cloth itself. The internal forces due to deformation of individual yarns include stretching and bending force, while the internal forces due to contact between interlaced yarns include normal compression, sliding friction, shearing force, and contact force between parallel yarns.

3.1 Gravity

For one single warp segment $[\boldsymbol{q}_0, \boldsymbol{q}_1]$ (similar for weft segment), we define the gravity potential energy as

$$V_{0,1} = \rho \Delta u \boldsymbol{g}^T \frac{\boldsymbol{x}_0 + \boldsymbol{x}_1}{2}. \tag{3}$$

We can derive the forces (Jacobian for energy) as

$$\begin{aligned}
\boldsymbol{F}_{\boldsymbol{x}_0} &= -\frac{\partial V}{\partial \boldsymbol{x}_0} = -\rho \Delta u \boldsymbol{g} \\
\boldsymbol{F}_{\boldsymbol{x}_1} &= \boldsymbol{F}_{\boldsymbol{x}_0} \\
\boldsymbol{F}_{u_1} &= -\frac{\partial V}{\partial u_1} = -\rho \boldsymbol{g}^T \frac{\boldsymbol{x}_0 + \boldsymbol{x}_1}{2} \\
\boldsymbol{F}_{u_0} &= -\boldsymbol{F}_{u_1}
\end{aligned}$$

We can derive the stiffness (Hessian for energy) as

$$\begin{aligned}
\frac{\partial \boldsymbol{F}_{\boldsymbol{x}_0}}{\partial u_1} &= -\rho \boldsymbol{g} \quad \frac{\partial \boldsymbol{F}_{\boldsymbol{x}_0}}{\partial u_0} = \rho \boldsymbol{g} \\
\frac{\partial \boldsymbol{F}_{\boldsymbol{x}_1}}{\partial u_1} &= -\rho \boldsymbol{g} \quad \frac{\partial \boldsymbol{F}_{\boldsymbol{x}_1}}{\partial u_0} = \rho \boldsymbol{g} \\
\frac{\partial \boldsymbol{F}_{u_0}}{\partial \boldsymbol{x}_0} &= \frac{1}{2} \rho \boldsymbol{g}^T \quad \frac{\partial \boldsymbol{F}_{u_0}}{\partial \boldsymbol{x}_1} = \frac{1}{2} \rho \boldsymbol{g}^T \\
\frac{\partial \boldsymbol{F}_{u_1}}{\partial \boldsymbol{x}_0} &= -\frac{1}{2} \rho \boldsymbol{g}^T \quad \frac{\partial \boldsymbol{F}_{u_1}}{\partial \boldsymbol{x}_1} = -\frac{1}{2} \rho \boldsymbol{g}^T
\end{aligned}$$

Generally, we assign the gravity $\boldsymbol{g}^T = (0, 0, -9.8)$.

3.2 Stretch

As for the stretch energy, we constrain that the actual length of segment equals to the arc length as much as possible. For one warp segment $[\boldsymbol{q}_0, \boldsymbol{q}_1]$ (similar for weft segment), we define the stretch energy as

$$V_{0,1} = \frac{1}{2} k_s \Delta u (||\boldsymbol{w}|| - 1)^2 \tag{4}$$

where $k_s = Y \pi R^2$, and Y is the elastic modulus. Yarns of woven cloth are often close to inextensible, which requires a high elastic modulus. As for the Jacobian and Hessian for stretch energy, it is all derived in the supplementary document of the original paper and I have checked its correctness. I recommend reader to refer that.

3.3 Bending

For the warp yarn shown in Figure 3, given an angle θ between segments $[\boldsymbol{q}_0, \boldsymbol{q}_2]$ and $[\boldsymbol{q}_0, \boldsymbol{q}_1]$, curvature density at node $\boldsymbol{q}_0$ is defined as $\kappa = \frac{2\theta}{u_1 - u_2}$.

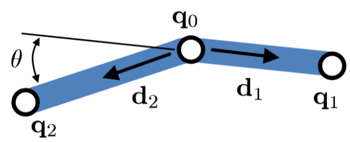

Figure 3: The bending case.

We define a bending energy density with stiffness k_b that is quadratic in curvature. Integrating it over the half-segments adjacent to $\boldsymbol{q}_0$ results in a discrete bending energy

$$V_{2,0,1} = k_b \frac{\theta^2}{u_1 - u_2} \tag{5}$$

The Jacobian and Hessian for bending energy can also be referred in the supplementary document. It should be noted that before we compute the bending force we should offset the node's position by yarn radius R along the normal direction, in case $\theta = 0$.

3.4 Compression at Yarn Crossings

As for the compression force at yarn crossings, we ignore the relative motion between warp and weft yarns along the normal direction, hence we cannot model normal compression as an elastic potential. After computing node's normal vector, we estimate the compression force by summing the normal components of stretch and bending forces $\boldsymbol{F}_s$ and $\boldsymbol{F}_b$ respectively, as shown in Figure 4, and averaging the resulting forces for both warp and weft directions, i.e.,

$$F_n = \frac{1}{2}\boldsymbol{n}^T(\boldsymbol{F}_s(u) + \boldsymbol{F}_b(u) - \boldsymbol{F}_s(v) - \boldsymbol{F}_b(v)) \tag{6}$$

And if the compression force is negative, we clamp the force to zero. It can be very complicated to compute the Jacobian for compression force directly because we get the normal vector by SVD decomposition. Thus we treat the normal vector to be constant while computing Jacobian for compression force and friction.

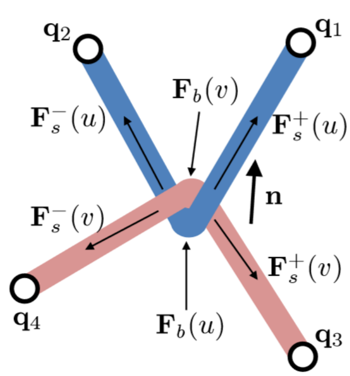

Figure 4: The compression force.

3.5 Friction

At each yarn crossing, the friction force can prevent sliding between warp and weft yarns. We model inter-yarn friction using a penalty-based approximation of the Coulomb model. It greatly simplifies the formulation of friction, and a simple spring on each sliding coordinate produces effective results.

When we initialize the cloth, we record the initial Eulerian position of each node, as anchor coordinates $\bar{u}$ and $\bar{v}$. We model friction as a zero-rest-length viscoelastic spring between the anchor position and current coordinate. The Coulomb model sets a limit μF_n to judge whether the node is in stick or slip mode. If the limit is not reached, it is in stick mode. And if the limit is exceeded, the node is in slip mode. We define the force as

$$F_{u_0} = \begin{cases} -k_f(u_0 = \bar{u}_0) & if\ stick \\ -sign(u_0 - \bar{u}_0)\mu F_n & if\ slip \end{cases} \tag{7}$$

In section 3.4 we have explained how to compute Jacobian for compression force, so it can be easy to get the Jacobians for friction force.

3.6 Shear

At yarn crossings, the adjacent warp and weft yarns will rotate on top of each other as a function of the shear angle ϕ, as shown in Figure 5. This rotation will produce two effects: yarn compression and contact friction.

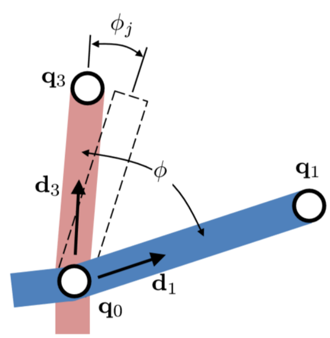

Figure 5: The shearing case.

To capture these effects, the angular friction force and elastic potential can be defined depending on the shear angle ϕ as follows

$$V_{0,1,3} = \frac{1}{2}k_x L(\phi - \frac{\pi}{2})^2 \tag{8}$$

Normal compression increases the resistance to shear, and we can model this effect through making the shear stiffness a function of the compression force, i.e., $k_x(F_n)$. In the original paper, the author also considered shear jamming and shear friction. However, in my implementation, I have not implemented them yet, while they will have little influence on the whole system's effect.

3.7 Contact between Parallel Yarns

Contact between adjacent parallel yarns can be easily modeled by adding a penalty energy if two yarn crossings get too close. Usually we define the distance threshold d as 4 times the yarn radius. We can define the contact force energy as

$$V_{0,1} = \begin{cases} \frac{1}{2}k_c L(u_1 - u_2 - d)^2 & if\ u_1 - u_2 < d \\ 0 & if\ u_1 - u_2 \geq d \end{cases} \tag{9}$$

where k_c is the penalty coefficient defined by myself. I assign $k_c = 1e7$ in the implementation. It is easy to compute the Jacobian and Hessian for contact energy and reader can refer to the supplementary documentation.

4 Implementation Details

I implemented the system in C++. The code is single-threaded currently and uses Eigen library for linear algebra and Mosek 8.1.0 for quadratic problems. I have uploaded the source code on my GitHub.

In my current implementation, I initialize the cloth as a set of nodes on a plane, and add a series of springs which stand for different kinds of forces. The crossing node inside the cloth has 5 DoFs while the node on the boundary of cloth is Lagrangian node with only 3 DoFs, which means the Eulerian coordinate for boundary node is constant. In fact, in real fabric weaving, the yarns are clamped on the end to prevent cloth from unraveling. In each simulation step, I firstly compute the generalized mass matrix and the normal vector for each node, and then compute the Jacobians and Hessians for each force spring one by one, filling the local matrix into the global stiffness matrix. After adding constraints, I solve the motion equation as a quadratic programming problem using Mosek 8.1.0, and then update each node's position and velocity. The system can be robust and stable when time step is assigned with 0.001s.

My current implementation has some subtle difference from original paper. As for shearing behavior, I regard the shear stiffness as a constant, and ignore the shear jamming and shear friction. Because these effects will have little influence on the cloth's dynamic kinematics.

5 Results

I render the dynamic simulation using OpenGL. And here is the rendering result. As for the animation effect, please see accompanying videos.

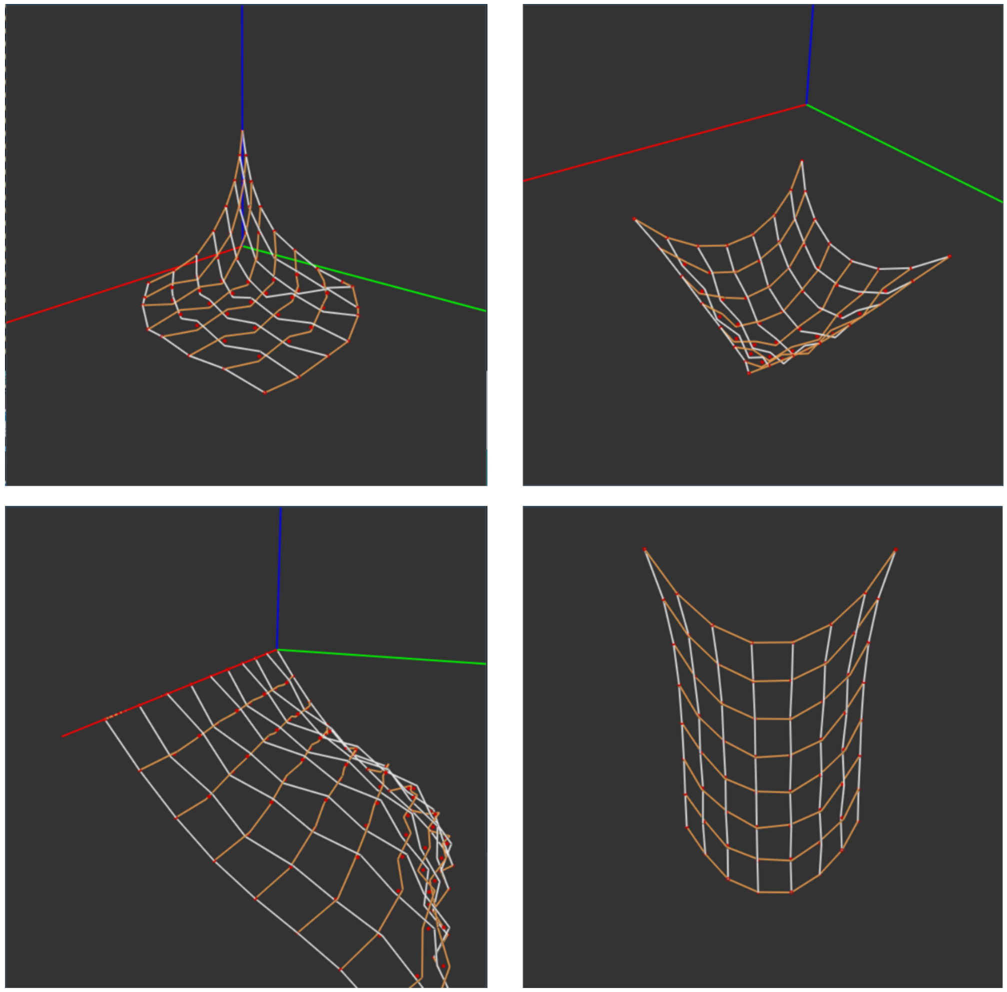

Figure 6: Results under different constraints.